

## JP10105208

Publication Title:

PATTERN RECOGNITION DEVICE AND ITS METHOD AND STORAGE MEDIUM STORING ITS PROGRAM

Abstract:

Abstract of JP10105208

PROBLEM TO BE SOLVED: To fast modify a sampling position to more accurately recognize a pattern by sampling patterns in a nonuniform manner and recognizing the original pattern based on an acquired signal. SOLUTION: A signal processor 2701 receives an input signal that is sampled by a two-dimensional array sensor 2705 through an optical system 2704, processes the input signal by using data stored in a storage device 2702 and sends a desired output to an output device 2703. It also sends a signal to control an input parameter (the orientation and position of an optical axis) of the system 2704 to a moving device 2708. The device 2708 is a device which makes a 1st housing 2706 perform two-dimensionally relative movement to a 2nd housing 2707, and two-dimensional movement amount is controlled by a control signal from the processor 2701. Even when the orientation where a recognition object pattern exists is not clear, the pattern can fast be acquired.

eof Data supplied from the esp@cenet database - Worldwide

-----  
Courtesy of <http://v3.espacenet.com>

特開平10-105208

(43)公開日 平成10年(1998) 4月24日

(51)Int.Cl.<sup>4</sup>  
G 0 5 B 19/00

識別記号

F I  
G 0 5 B 19/00

審査請求 未請求 請求項の数27 ○L (全 39 頁)

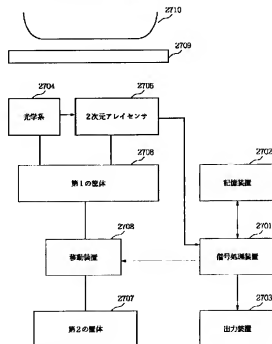
(21)出願番号 特願平8-258743  
(22)出願日 平成8年(1996) 9月30日(71)出願人 000001007  
キヤノン株式会社  
東京都大田区下丸子3丁目30番2号  
(72)発明者 鷺澤 舞芳  
東京都大田区下丸子3丁目30番2号キヤノ  
ン株式会社内  
(74)代理人 弁理士 丸島 儀一

(54)【発明の名称】 パターン認識装置及びその方法、及びそのプログラムを記憶した記憶媒体

(57)【要約】

【課題】 2次元パターンの認識を、簡単なハードウェア構成により少ない処理時間で達成する。

【解決手段】 パターン認識装置に、認識対象のパターンが描かれた原稿2710からの光線を結像させる光学系2704と、結像した画像をサンプリングする2次元アレイセンサ2705と、このサンプリングによって得られた多重解像度部分画像に基づいて、原稿2710に描かれたパターンを認識する信号処理装置2701と、信号処理装置2701からの制御信号に基づいて、原稿2710に対して光学系2704の光軸を移動させる移動装置2708とを設ける。



## 【特許請求の範囲】

【請求項1】 認識対象のパターンを表わす第1の信号を入力する入力手段と、

該入力手段より入力された第1の信号を非一様にサンプリングするサンプリング手段と、

該サンプリング手段のサンプリングによって得られた第2の信号に基づいて、前記第1の信号の表わすパターンを認識する認識手段と、

該認識手段による認識結果に基づいて、前記サンプリング手段による非一様サンプリングのサンプリング位置を変更する変更手段とを有することを特徴とするパターン認識装置。

【請求項2】 前記第1の信号は2次元の画像信号であることを特徴とする請求項1に記載のパターン認識装置。

【請求項3】 前記サンプリング手段は、前記第2の信号として、ある注視点近傍の多重解像度画像を抽出することを特徴とする請求項2に記載のパターン認識装置。

【請求項4】 前記サンプリング手段は、前記多重解像度画像の各解像度の各画素の値を、各解像度に対応した大きさの範囲の入力画像を用いて算出することを特徴とする請求項3に記載のパターン認識装置。

【請求項5】 前記サンプリング手段において、前記範囲の大きさを高解像度側と大きな範囲とすることを特徴とする請求項4に記載のパターン認識装置。

【請求項6】 前記変更手段が、前記認識結果に基づいて、前記多重解像度画像における異なる解像度の画像間の相互情報量を算出する相互情報量算出手段と、前記相互情報量に基づいて前記注視点を変更する注視点変更手段とを有することを特徴とする請求項2に記載のパターン認識装置。

【請求項7】 前記認識手段が、前記多重解像度画像に対応する確率セルオートマトンについて、パターンに基づく各セルの状態値の確率分布を算出する確率分布算出手段と、当該確率分布に基づいて、前記第1の信号の表わすパターンを特定する同定手段とを有することを特徴とする請求項2に記載のパターン認識装置。

【請求項8】 前記確率分布算出手段は、前記確率セルオートマトンにおける注目するセルの状態値の確率分布を、該注目するセルより低解像度の近傍のセルの状態値の確率分布に基づいて算出することを特徴とする請求項7に記載のパターン認識装置。

【請求項9】 前記相互情報量算出手段は、前記確率セルオートマトンにおける注目する解像度のセルの状態値の確率分布と該注目する解像度のセルより高解像度のセルの状態値の確率分布とに基づいて相互情報量を算出することを特徴とする請求項7に記載のパターン認識装置。

【請求項10】 セルの状態値と、画像パターンまたは下位のセルの状態値のパターンとの対応関係を記憶する

対応関係記憶手段を具え、当該対応関係を参照して前記同定手段がパターンを特定することを特徴とする請求項7に記載のパターン認識装置。

【請求項11】 前記対応関係記憶手段に記憶すべき対応関係を学習する学習手段を備えることを特徴とする請求項10に記載のパターン認識装置。

【請求項12】 注目するセルの状態値の確率分布のエントロピーを算出するエントロピー算出手段を備え、該エントロピー算出手段により算出されたエントロピーが所定の条件を満たした場合に前記同定手段による処理を実行することを特徴とする請求項7に記載のパターン認識装置。

【請求項13】 前記入力手段が画像信号を光学的に入力する光学の入力手段を備え、前記変更手段が当該光学の入力手段の光軸を移動させる移動手段を備えることを特徴とする請求項2に記載のパターン認識装置。

【請求項14】 認識対象のパターンを表わす第1の信号を入力する入力工程と、

該入力工程により入力された第1の信号を非一様にサンプリングするサンプリング工程と、

該サンプリング工程のサンプリングによって得られた第2の信号に基づいて、前記第1の信号の表わすパターンを認識する認識工程と、

該認識工程による認識結果に基づいて、前記サンプリング工程による非一様サンプリングのサンプリング位置を変更する変更工程とを有することを特徴とするパターン認識方法。

【請求項15】 前記第1の信号は2次元の画像信号であることを特徴とする請求項14に記載のパターン認識方法。

【請求項16】 前記サンプリング工程は、前記第2の信号として、ある注視点近傍の多重解像度画像を抽出することを特徴とする請求項15に記載のパターン認識方法。

【請求項17】 前記サンプリング工程は、前記多重解像度画像の各解像度の各画素の値を、各解像度に対応した大きさの範囲の入力画像を用いて算出することを特徴とする請求項16に記載のパターン認識方法。

【請求項18】 前記サンプリング工程において、前記範囲の大きさを高解像度側と大きな範囲とすることを特徴とする請求項17に記載のパターン認識方法。

【請求項19】 前記変更工程が、前記認識結果に基づいて、前記多重解像度画像における異なる解像度の画像間の相互情報量を算出する相互情報量算出工程と、前記相互情報量に基づいて前記注視点を変更する注視点変更工程とを有することを特徴とする請求項15に記載のパターン認識方法。

【請求項20】 前記認識工程が、前記多重解像度画像に対応する確率セルオートマトンについて、パターンに基づく各セルの状態値の確率分布を算出する確率分布算

出工程と、当該確率分布に基づいて、前記第1の信号の表わすパターンを特定する特定工程とを有することを特徴とする請求項1に記載のパターン認識方法。

【請求項21】 前記確率分布算出工程では、前記確率セルオートマトンにおける注目するセルの状態値の確率分布と、該注目するセルより低解像度の近傍のセルの状態値の確率分布に基づいて算出することを特徴とする請求項20に記載のパターン認識方法。

【請求項22】 前記相互情報量算出工程は、前記確率セルオートマトンにおける注目する解像度のセルの状態値の確率分布と該注目する解像度のセルより高解像度のセルの状態値の確率分布とに基づいて相互情報量を算出することを特徴とする請求項20に記載のパターン認識方法。

【請求項23】 前記特定工程が、対応関係メモリに記憶された、セルの状態値と、画像パターンまたは下位のセルの状態値のパターンとの対応関係を参照して、パターンを特定することを特徴とする請求項20に記載のパターン認識方法。

【請求項24】 前記対応関係メモリに記憶すべき対応関係を学習する学習工程を備えることを特徴とする請求項23に記載のパターン認識方法。

【請求項25】 注目するセルの状態値の確率分布のエントロピーを算出するエントロピー算出工程を備え、該エントロピー算出工程より算出されたエントロピーが所定の条件を満たした場合に前記特定工程の処理を実行することを特徴とする請求項20に記載のパターン認識方法。

【請求項26】 前記入力工程では、光学的入力部を用いて画像信号を光学的に入力し、前記変更工程が当該光学的入力部の光軸を移動させる移動工程を備えることを特徴とする請求項15に記載のパターン認識装置。

【請求項27】 認識対象のパターンを表わす第1の信号を入力する入力工程と、該入力工程により入力された第1の信号を非一様にサンプリングするサンプリング工程と、該サンプリング工程のサンプリングによって得られた第2の信号に基づいて、前記第1の信号の表わすパターンを認識する認識工程と、該認識工程による認識結果に基づいて、前記サンプリング工程による非一様サンプリングのサンプリング位置を変更する変更工程とを有することを特徴とするパターン認識プログラムを記憶した記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、パターン認識装置に関するものであり、特に、パターン認識装置に含まれる入力装置が、ある種の制約を受け、その制約をパラメタによって変化させることが可能であるときに、効率的にパターンを認識するためのパラメタを制御する

パターン認識装置に関する。

【0002】

【従来の技術】 例えば、3次元空間を移動するロボットに設けられた画像認識装置では、認識すべき画像信号が周囲のどの方位に存在するかが予め分かっていないので、できるだけ広い範囲の信号を入力することが要求される。しかし同時に、あるパターンを認識するために十分なだけの空間解像度をも備えていなければならない。これらを同時に満足するような画像入力装置として、人間の網膜特性を参考にして、光軸の中心付近を高解像度で、光軸から離れるに従って低解像度になるような非一様サンプリングが考案された。

【0003】 このような非一様サンプリングでは、周辺の低解像度でサンプリングされたパターンを正確に認識するためには、そのパターンが光軸付近の高解像度領域でサンプリングされるように光軸を変更しなければならない。即ち、非一様サンプリングは光軸制御を伴って初めて効果的な入力方式となる。

【0004】 そのための光軸制御方式としては、非一様サンプリングによって入力された入力画像の特徴量に基づいて制御するものが考案されている。この特徴量としては、例えば画像強度の傾きの絶対値、ある2次元フィルタを施したあとの出力値、等が用いられている。また、知識データベースに基づく光軸制御方式も考案されている。この知識データベース方式では、システムが遂行すべきタスクや、認識対象の画像或いはパターンの集合と、ベイズネットワークで表現し、この確率構造に基づいて、与えられたタスクを効率よく遂行するために次に行うべき動作を決定している。

【0005】

【発明が解決しようとする課題】 上述したように、従来における光軸制御の殆どは、画像信号に基づく方式であり、従ってエッジ強度が強い箇所やフィルタ出力の強い箇所という画像の幾何学的性質のみで光軸が制御されてしまう。このような制御方式では、タスクに必要でない箇所でも、傾きの絶対値が大きければそこに光軸を移動させてしまう。

【0006】 また、上述の知識データベース方式では、ベイズネットワークを用いているために、表現できるデータの構造が限定されてしまう。

【0007】

【課題を解決するための手段】 上記課題を解決するために、本発明によれば、パターン認識装置に、認識対象のパターンを表わす第1の信号を入力する入力手段と、該入力手段より入力された第1の信号を非一様にサンプリングするサンプリング手段と、該サンプリング手段のサンプリングによって得られた第2の信号に基づいて、前記第1の信号の表わすパターンを認識する認識手段と、該認識手段による認識結果に基づいて、前記サンプリング手段による非一様サンプリングのサンプリング位置を

変更する変更手段とを備える。

【0008】本発明の他の態様によれば、パターン認識方法に、認識対象のパターンを表わす第1の信号を入力する入力工程と、該入力工程により入力された第1の信号を非一様にサンプリングするサンプリング工程と、該サンプリング工程のサンプリングによって得られた第2の信号に基づいて、前記第1の信号の表わすパターンを認識する認識工程と、該認識工程による認識結果に基づいて、前記サンプリング工程による非一様サンプリングのサンプリング位置を変更する変更工程とを備える。

【0009】本発明の他の態様によれば、記憶媒体に、認識対象のパターンを表わす第1の信号を入力する入力工程と、該入力工程により入力された第1の信号を非一様にサンプリングするサンプリング工程と、該サンプリング工程のサンプリングによって得られた第2の信号に基づいて、前記第1の信号の表わすパターンを認識する認識工程と、該認識工程による認識結果に基づいて、前記サンプリング工程による非一様サンプリングのサンプリング位置を変更する変更工程とを備えるパターン認識プログラムを記憶する。

$$A(1, m, n, *) = (A(1, m, n, 1), A(1, m, n, 2), \dots, A(1, m, n, W))^T \quad (1)$$

【0014】

$$A(1, *, w) = \begin{pmatrix} A(1, 1, 1, w) & A(1, 1, 2, w) & \dots & A(1, 1, N, w) \\ A(1, 2, 1, w) & A(1, 2, 2, w) & \dots & A(1, 2, N, w) \\ \dots & \dots & \dots & \dots \\ A(1, M, 1, w) & A(1, M, 2, w) & \dots & A(1, M, N, w) \end{pmatrix} \quad (2)$$

【0015】更に、インデクスに「が現れたときには、その全てのインデクスに渡る値の集合を表すものとす

$$A(1, m, n, \sim) = \{A(1, m, n, 1), A(1, m, n, 2), \dots, A(1, m, n, W)\} \quad (3)$$

なお、～は複数個のインデクスに現れてもよい。

【0016】また確率理論に基づいて説明を行うので、有限個の非負の要素で構成される有限次元ベクトル $(z(1), z(2), \dots, z(N))^T$ に対して、総和が1になる

$$\lambda(z) = (z(1)/Z, z(2)/Z, \dots, z(N)/Z)^T \quad \text{if } Z \neq 0 \\ - (1/N, 1/N, \dots, 1/N)^T \quad \text{otherwise} \quad (4)$$

ただし、Zはベクトルの要素の総和である：

【0018】

$$Z = \sum_{n=1}^{n=N} z(n) \quad (5)$$

【0019】以下に、図1のパターン認識装置の各部を説明する。

【0020】＜入力装置101＞入力装置101

は、認識の対象とする信号を受信し、それを信号処理装

【0010】

【発明の実施の形態】

（実施形態1）図1は、本発明の1実施形態であるパターン認識装置の機能構成を表す図である。実施形態1は、オフラインパターン認識への応用例である。装置の各部は、以下に説明する機能を実現するものであれば、それぞれが専用のプログラムメモリやプロセッサを含む機器であってもよいし、複数の機能部を、同一のCPUがROMやディスクメモリ等に記憶された各機能プログラムを実行することにより、あるいは各機能に対応する特定のハードウェアを制御する制御プログラムを実行することにより実現してもよい。

【0011】ここで、以下の記述を簡単にするため次のような記述を採用する。

【0012】まず、インデクスにうちの1つに\*が現れたときには、その全てのインデクスに対する値を要素とするベクトルを表し、2つに\*が現れたときには、それら全てのインデクスに対する値を要素とする行列を表すものとする。

【0013】例えば：

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

【外1】

</

としての信号であり、この信号は、音声信号のような1次元信号でも、画像のような2次元信号でもよい。本実施形態では、入力信号を2次元信号 $I(*,*)$ とする。

【0022】<信号処理装置102>信号処理装置102は、通常のモードでは、入出力装置101から送信された2次元信号を認識し、学習モードのときには、入出力装置101から送信された2次元信号と教師信号とを基に、認識のための知識データを、後述する量子化コードブックなどの形で記憶装置103に格納する。

【0023】図2に、信号処理装置102の処理のフローチャートを示す。

【0024】ステップS201では、入出力装置101から入力があったかどうかを検査する。入力がない場合は、ステップS201を繰り返し、入力があった場合には、ステップS202に進む。ステップS202では、学習モードかどうかを検査し、学習モードであればステ

$$A = \{Z^4, Q, X, \phi\}$$

【0028】本実施形態では階層構造を考えているので、セル空間 $Z^4$ は、 $\phi=3$ 、即ち $(l, m, n) \in Z^3$ 、ここで、 $l$ は階層レベル、 $(m, n)$ は2次元画像に対応する2次元空間位置を表している。セルの状態の集合は、状態値が可算個の場合には、一般に自然数の集合と考えるとよい。更に状態値を有限個に制限すれば、その上の確率分布

$$X(l, m, n) = \{(1, 1, 3m-1, 3n-1), (1, 1, 3m-1, n), (1, 1, 3m-1, 3n+1), \\ (1, 1, 3m, 3n-1), (1, 1, 3m, 3n), (1, 1, 3m, 3n+1), \\ (1, 1, 3m+1, 3n-1), (1, 1, 3m+1, 3n), (1, 1, 3m+1, 3n+1)\} \quad (7)$$

のように定義する。例えば、座標 $(2, 1, 1)$ のセル803

$$X(2, 1, 1) = \{(1, 2, 2), (1, 2, 3), (1, 2, 4), (1, 3, 2), (1, 3, 3), \\ (1, 3, 4), (1, 4, 2), (1, 4, 3), (1, 4, 4)\} \quad (8)$$

となる。

【0030】写像 $\phi$ は、セル空間上で $(l, m, n)$ に位置するセルの状態値 $\mu$ が $v$ となる確率 $P_{l, m, n}(\mu=v)$ 、(7)式で定義されている近傍系に含まれるセル $(i, j, k)$ の状態値 $\omega_{i, j, k}$ が $w$ となる確率 $P_{i, j, k}(\omega=w)$ 、及びこれらの間の

$$P_{l, m, n}(\mu=v) = \phi \{ P_{i, j, k}(\omega=w), (i, j, k) \in X(l, m, n) \}$$

$$= \prod_{(i, j, k) \in X(l, m, n)} \sum_{w \in \Omega} \pi_{i, j, k}(\mu=v | \omega=w) P_{i, j, k}(\omega=w) \quad (9)$$

【0032】図8に、 $l=(1, 2, 3)$ 、 $m=(-1, 0, 1)$ 、 $n=(-1, 0, 1)$ の場合の確率セルオートマトンAの構造を示す。図中、鉛直上向きに階層レベル $l$ 、水平右方向に空間座標 $m$ 、右斜め上方に空間座標 $n$ が取られている。図中、セル

$$X(3, 0, 0) = \{(2, -1, -1), (2, -1, 0), (2, -1, 1), (2, 0, -1), \\ (2, 0, 0), (2, 0, 1), (2, 1, -1), (2, 1, 0), (2, 1, 1)\} \quad (10)$$

である。

【0033】以下の説明では、解像度レベル $l$ 、2次元空

$$l = \{1, 2, \dots, L\} \\ m = \{-M_0, -M_0+1, \dots, 0, \dots, M_0-1, M_0\} \\ n = \{-N_0, -N_0+1, \dots, 0, \dots, N_0-1, N_0\}$$

間 $S204$ へ、そうでなければステップS203へ進む。

【0025】ステップS203では、認識処理を実行する。この認識処理は図3及び図4につき後述する。ステップS204では、教師信号が入力されたかどうかを検査し、入力されたならばステップS205へ進み、学習処理を行なう。入力されなければステップS204の処理を繰り返す。ステップS205で実行される学習処理は図4につき後述する。

【0026】次に、上述したステップS203で実行される認識処理の流れを、図3～4のフローチャートに沿って説明する。

【0027】まず、ステップS301で確率セルオートマトンAを初期化する。ここで、確率セルオートマトンAは、セル空間 $Z^4$ 、各セルの状態の集合 $\Omega$ の上に定義された確率空間 $\pi$ 、近傍系 $X$ 、写像 $\phi$ の4つ組で表される：

$$(6)$$

は状態値1に対する確率、2に対する確率、…というように与えられ、これらを1まとまりとして、有限次元ベクトルとして表現できる。

【0029】また、セル空間での座標 $(l, m, n)$ に位置するセルに対する近傍系 $X(l, m, n)$ は、例えば

条件付き確率 $\pi_{i, j, k}(\mu=v | \omega=w)$ を用いて次式で与えられる：

$$【0031】$$

【外4】

801のセル空間での座標は $(3, 1, 1)$ 、セル802の座標は $(2, 1, 0)$ である。また近傍系は実線で示しており、例えばセル $(3, 0, 0)$ の近傍系 $X(3, 0, 0)$ は

$$w = \{1, 2, \dots, W\}$$

(11)

とする。

【0034】(11)式のインデクスを用いれば、階層レベ

ル $l$ 、2次元空間位置 $(m, n)$ に位置するセルの状態値 $\omega$ が $w$ を取る確率 $P_{l,m,n}(\omega=w)$ は

$$P_{l,m,n}(\omega=w) = \Lambda(1, m, n, w) \quad (12)$$

【0035】更に、セル $(1, m, n)$ の状態値の確率分布は、 $W$ 次元ベクトル $A(1, m, n, *)$ として記述できる。 $n, W)^T$ ),また写像 $\phi$ は次のように定義し直せる:

【0036】

【外5】

$$A(1, m, n, *) = \lambda \{ A(1, m, n, 1), A(1, m, n, 2), \dots, A(1, m,$$

$$A(1, m, n, w) = \phi(\{A(i, j, k, *), (i, j, k) \in X(1, m, n)\})$$

$$= \prod_{(i,j,k) \in X(1,m,n)} \langle \phi(1, m, n, w | i, j, k, *) | A(i, j, k, *) \rangle \quad (13)$$

ここで、 $\langle a | b \rangle$ はベクトル $a$ と $b$ の内積、 $\psi(1, m, n, w |$  $i, j, k, *)$ は次式

$$\psi(1, m, n, w | i, j, k, *) = (\psi(1, m, n, w | i, j, k, 1), \dots, \psi(1, m, n, w | i, j, k, W))^T \quad (14)$$

で定義される $W$ 次元ベクトルである。ここで、 $\psi(1, m, n, w | i, j, k, v)$ は、セル $(i, j, k)$ の状態値が $v$ のときにセル $(1, m, n)$ の状態値が $w$ となる条件付き確率である。【0037】図20に、写像 $\psi(1, m, n, w | i, j, k, v)$ の例を示す。図中、縦横それぞれ3個のマスの目で表現されているのが、セル $(1, m, n)$ に対する近傍系 $X(1, m, n)$ である。それぞれのマス目の下に書かれているのが、セル $(1, m, n)$ の状態値 $w$ である。それぞれのマス目の中の数字

$$\psi(1, m, n, w | i-1, 1, 1, *) = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, \dots, 0)^T$$

(15)

となる。マス目のインデクスを通常の行列の表記に促え

ば、(13)式は実際に以下のように計算できる:

$$\begin{aligned} A(1, m, n, w) &= \psi(1, m, n, w | i-1, 1, 1, *) A(i-1, 1, 1, *)^T \\ &\times \psi(1, m, n, w | i-1, 1, 2, *) A(i-1, 1, 2, *)^T \\ &\times \psi(1, m, n, w | i-1, 1, 3, *) A(i-1, 1, 3, *)^T \\ &\times \psi(1, m, n, w | i-1, 2, 1, *) A(i-1, 2, 1, *)^T \\ &\times \psi(1, m, n, w | i-1, 2, 2, *) A(i-1, 2, 2, *)^T \\ &\times \psi(1, m, n, w | i-1, 2, 3, *) A(i-1, 2, 3, *)^T \\ &\times \psi(1, m, n, w | i-1, 3, 1, *) A(i-1, 3, 1, *)^T \\ &\times \psi(1, m, n, w | i-1, 3, 2, *) A(i-1, 3, 2, *)^T \\ &\times \psi(1, m, n, w | i-1, 3, 3, *) A(i-1, 3, 3, *)^T \end{aligned} \quad (16)$$

【0039】図3に戻り、ステップS302で、入力画像1における注視点座標 $g=(gx, gy)$ を $g_{\text{int}}=(gx_{\text{int}}, gy_{\text{int}})$ に初期化する。

【0040】ここで、注視点とは、入力画像1から後述する多重解像度画像Dを抽出する際の中心点である。注視点は、多重解像度画像と対応付けるため、多重解像度

表現で表される。本実施形態で用いる多重解像度画像は、スケールが3のべき乗で変化しているため、注視点の座標も3のべき乗、即ち3進数で記述する:

【0041】

【外6】

$$gx = gx_{\text{int}} + \sum_{j=0}^{l-1} gx_j 3^j, \quad gy = gy_{\text{int}} + \sum_{j=0}^{l-1} gy_j 3^j \quad (17)$$

【0042】ここで、 $gx_j$ 及び $gy_j$ の取り得る値は $\{1, 2, 3\}$  或いは $\{1, 0, 1\}$ である。多重解像度画像のスケール変換が2のべき乗に基づいている場合には、注視点の座標は2進数で記述することになる。図25に、(17)式による記述と、入力画像上の座標との関係を示す。【0043】次に、ステップS303では、入力画像から注視点座標近傍の多重解像度画像Dを抽出する。この処理は、図4から図7につき後述する。多重解像度画像Dの要素は、解像度レベルを表す添え字 $l$ と、注視点座標に対して相対的な空間位置を表す2つの添え字 $(m, n)$ を

用いて $D(l, m, n)$ と表される。図9の黒く塗られた部分  
が、 $l=\{0, 1, 2\}$ 、 $m=\{-1, 0, 1\}$ 、 $n=\{-1, 0, 1\}$ のときの多重  
解像度画像Dの例である。以下の説明では、確率セルオ

$$\begin{aligned} l &= \{0, 1, \dots, L-1\} \\ m &= \{-N_0, -N_0+1, \dots, 0, \dots, N_0-1, N_0\} \\ n &= \{-N_0, -N_0+1, \dots, 0, \dots, N_0-1, N_0\} \end{aligned} \quad (18)$$

とする。

【0044】上式及び図9からわかるように、本実施形  
態で述べている多重解像度画像は、通常の多重解像度画  
像の部分画像となっている。図中、901が解像度レベ  
ル2に属する画像、902が解像度レベル1に属する画  
像、903が解像度レベル0に属する画像である。また  
レベル1の9画素から成る画像の占める空間領域と、レ  
ベル1+1の画像のうち中心に位置する画素の占める空間  
領域とは一致する。

【0045】多重解像度画像の構成方法はいくつかあ

$$\begin{aligned} l &= \{1, \dots, L-1\} \\ m &= \{-N_0, -N_0+1, \dots, 0, \dots, N_0-1, N_0\} \\ n &= \{-N_0, -N_0+1, \dots, 0, \dots, N_0-1, N_0\} \end{aligned} \quad (19)$$

である。以下、場合によって、MD及びNDを1、またMB  
及びNBを2として説明する。

【0047】次に、ステップS305で、 $l$ を1に設定  
する。

【0048】ステップS306では、確率セルオートマ  
トンのセルのうち、レベル $l$ に属するセル $A(l, \sim, \sim)$   
(本実施形態では9画素)の状態値の確率分布(次元ベ  
クトル) $A(l, \sim, \sim, \cdot)$ を更新する。この確率分布 $A(l, \sim, \sim, \cdot)$   
の更新処理は図10から図12につき後述する。

【0049】ステップS307では、確率セルオートマ  
トンAのレベル $l$ のセルとレベル $l+1$ のセルの間の相互情

$$\varepsilon = - \sum_{w=1}^{w=W} A(l, n, n, w) \log A(l, m, n, w) \quad (20)$$

【0053】ステップS311では、(20)式で計算した  
 $\varepsilon$ が予め定めた値 $\varepsilon_c$ より小さいかどうかを検査し、小  
さくなければステップS312へ進む。小さければ認識  
処理を終了し、この時の状態値の確率分布から、認識し  
たいセルの状態値を確率が最大となる状態値に決定す  
る。この状態値から、後述する対応表または量子化コード  
ブックの対応関係を参照して、対応する画像を認識結  
果として出力する。

【0054】ステップS312では、多重解像度データ  
Bを用いて抽出画像の人力画像における注視点座標を更  
新する。この処理は図18につき後述する。

【0055】次に、ステップS303で実行する処理を  
図4から図7に沿って説明する。

【0056】まず、ステップS501で $l$ を1に設定す  
る。

【0057】続いて、ステップS502では、注視点 $s=(x_s, y_s)$ における解像度レベル $l$ の画像

ートマトンの場合と同様、解像度レベル $l$ 、2次元空間位  
置 $m$ 及び $n$ の範囲をそれぞれ、

て、例えば、入力画像に対して各画素の占める空間領域  
に渡る平均値を画素値とするもの、ウェーブレット変換  
のように空間スケールの異なる積分核を用いて得られた  
積分変換係数を画素値とするもの等を利用することがで  
きる。図9では、積分核904及び905を用いる。

【0046】ステップS304では、多重解像度データ  
B及びCを初期化する。ここでB及びCの要素は上記  
と同様、 $B(l, m, n)$ 及び $C(l, m, n)$ のように3つの添え字  
で表され、添え字の範囲はそれぞれ、

報量を算出しBに格納する。相互情報量算出処理は図1  
4から図16につき後述する。

【0050】ステップS308では、 $l$ を $l+1$ に更新す  
る。ステップS309では、レベル $l$ が $l$ を超えたかど  
うかを検査し、超えていたらステップS310に進む。  
そうでなければステップS305に進む。

【0051】ステップS310では、認識したいセルの  
状態値 $\omega$ の確率分布のエントロピー $\varepsilon$ を以下の式により算  
出する。

【0052】  
【外7】

$D(l, m, n), -N_0 \leq m \leq N_0, \dots, N_0 \leq n \leq N_0$   
を入力画像から抽出する。この処理は図6及び図7につ  
き後述する。

【0058】ステップS503では $l$ を $l+1$ に更新する。  
ステップS504で $l$ が $l$ を超えたかどうかを検査し、  
超えていれば処理を終了し、そうでなければステップS  
502に戻る。

【0059】以上により、注視点における解像度レベル  
1から解像度レベル $L$ までの画像を入力画像から抽出す  
る。

【0060】次に、ステップS502で実行する、入力  
画像から注視点における解像度レベル1の画像を抽出す  
る処理を図6及び図7に沿って説明する。

【0061】ステップS601で、解像度レベルの値か  
ら、抽出画像の1画素 $D(l, m, n)$ に対する入力画像1におけ  
る受容野の大きさを算出する。

【0062】ここで受容野とは、抽出画像の1画素の値



を計算するために利用する入力画像の画素の範囲のことであり、本実施形態では縦横3画素の正方形格子を用いている。即ち、レベル $l=0$ のときは縦横1画素、レベル $l=1$ のときは縦横3画素というようになる。以下、受容野の範囲を縦横それぞれ、 $-\Delta x$ から $\Delta x$ 、 $-\Delta y$ から $\Delta y$ と書く。

【0063】ステップS602で $y=-N_0$ に設定する。

$$D(l, x, y) = \sum_{dy=-\Delta y}^{dy=\Delta y} \sum_{dx=-\Delta x}^{dx=\Delta x} \theta(dx, dy) I(gx+x+dx, gy+y+dy) \quad (21)$$

上式は受容野に渡る重み付き平均値の計算式である。

【0066】ステップS605で $x$ を $x+1$ に更新する。ステップS606で $x$ が $N_0$ を超えたかどうかを検査し、超えていなければステップS607へ進み、そうでなければステップS604へ戻る。

【0067】ステップS607で $y$ を $y+1$ に更新する。ステップS608で $y$ が $N_0$ を超えたかどうかを検査し、超えていなければ処理を終了し、そうでなければステップS603へ戻る。

【0068】ステップS306での確率分布の更新処理を、図10から図12のフローチャート、及び図13に沿って説明する。

【0069】図13で、1301が入力画像1、1302が多重解像度画像D、1303が確率セルオートマトンAである。図は簡単のために1次元的に表現している。

【0070】多重解像度画像Dは解像度レベルが下から0、1、2、それぞれの解像度レベルに縦横3画素、合計9画素よりなる画像がある。

【0071】確率セルオートマトンAの階層レベルは、

$$\alpha = \lambda((\alpha_1, \alpha_2, \dots, \alpha_w)^T) \quad (22)$$

【0075】第1の対応関係1303がベクトル量子化のコードブックのときには、入力画像D(0,\*,\*)と、コードブックに格納されている代表ベクトルとの内積を計算し、符号語に対する内積の値を $\alpha$ とする。

【0076】ステップS1003では、 $\alpha$ の要素に最大値が存在するかどうかを検査し、存在すればステップS

$$A(l, sx_1, sy_1, kmax) = 1, \quad (23)$$

$$A(l, sx_1, sy_1, k) = 0, \quad k \neq kmax \quad (24)$$

【0078】ステップS1005で、後述する確率分布ベクトル $\beta$ の値を階層レベル $l$ の注視点に対応するセル

$$A(l, sx_1, sy_1, *) = \beta \quad (25)$$

【0079】ステップS1006で $m=-N_0$ に設定する。ステップS1007で $n=-N_0$ に設定する。

【0080】ステップS1008では、次式により、多

$$dP = (dP_1, dP_2, \dots, dP_N)^T \quad (26)$$

$D(l, sx_1+m, sy_1+n)$ が離散値 $\{1, 2, \dots, N\}$ を取るとき、上式は変換行列 $P$ を用いて次のように書き直せる：

ステップS603で $x=-N_0$ に設定する。

【0064】ステップS604で、抽出画像のレベル1、2次元格子位置が $(x, y)$ の1画素の値を次式で算出する：

【0065】

【外8】

1、2、3の3階層であり、それぞれの階層レベルに縦横3個、合計9個のセルが存在する。多重解像度画像Dは、注視点を中心として入力画像から抽出される。解像度レベル0の9個の画素値は入力画像と一致する。解像度レベル1に属する9個の画素値は、入力画像の縦横3個、合計9個の画素を適当な積分核1304、例えば図9の905を施して、積分変換することによって得られる。同様に、解像度レベル2に属する9個の画素値は、入力画像の縦横9個、合計81個の画素を適当な積分核1305、例えば図9の904を施して、積分変換することによって得られる。

【0072】次に、確率分布の更新処理手順を説明する。

【0073】まず、ステップS1001で、階層レベル1が0かどうかを検査し、そうであればステップS1002に、そうでなければステップS1005へ進む。

【0074】ステップS1002では、解像度レベル0の画像D(0,\*,\*)1302と第1の対応関係1308を用いて、次式により $\alpha$ を算出する：

1004へ、存在しなければステップS1006へ進む。

【0077】ステップS1004では、 $\alpha$ の要素のうち最大値を取る要素を $kmax$ とすると、次式で確率セルオートマトンの状態値の確率分布を更新する：

の状態値の確率分布として代入する：

重解像度画像D(1,  $sx_1+m, sy_1+n$ )の値に対する条件付き確率分布ベクトル $dP$ を算出する：

【0081】

【外9】

$$dP = \begin{pmatrix} f(1,1) & f(1,2) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \dots & \dots & \dots & \dots \\ f(W,1) & f(W,2) & \dots & f(W,N) \end{pmatrix} \begin{pmatrix} \Pr(D(1, gx_1+m, gy_1+n)=1) \\ \Pr(D(1, gx_1+m, gy_1+n)=2) \\ \dots \\ \Pr(D(1, gx_1+m, gy_1+n)=N) \end{pmatrix}$$

ただし、 $\Pr(D(1, gx_1+m, gy_1+n)=k)$ は、 $D(1, gx_1+m, gy_1+n)$ が $k$ である確率である。従って、 $D(1, gx_1+m, gy_1+n)$ が2のときには、 $dP$ は $(f(1,2), f(2,2), \dots, f(W,2))^T$ と

$$A_{NEW}(1, gx_1+m, gy_1+n, *) = \lambda ((A_{NEW}(1), A_{NEW}(2), \dots, A_{NEW}(W))^T), \\ A_{NEW}(j) = A_{OLD}(1, gx_1+m, gy_1+n, j) \quad dP_j, \quad \forall j \in \{1, 2, \dots, W\} \quad (27)$$

【0083】ステップS1010で $n$ を $n+1$ に更新する。ステップS1011で $n$ が $N_0$ を超えているかどうかを検査し、超えていればステップS1012へ、そうでなければステップS1008へ進む。

【0084】ステップS1012では $m$ を $m+1$ に更新する。ステップS1013で $m$ が $M_0$ を超えているかどうかを検査し、超えていれば処理を終了し、そうでなければステップS1007へ戻る。

【0085】次に、図4のステップS307で実行される相互情報量の算出処理を、図14から図16のフローチャートを参照しながら説明する。また、図17に、この処理における各データ間の関係を図示する。

【0086】まず、ステップS1401で、確率分布ベクトル $\beta$ を零ベクトルに初期化する。この確率分布ベクトル $\beta$ は、例えば図17の中で、確率セルオートマトン

$$X(1+1, gx_1+m, gy_1+n) = \{(1, gx_1+m-1, gy_1+n-1), (1, gx_1+m-1, gy_1+n), \\ (1, gx_1+m-1, gy_1+n+1), (1, gx_1+m, gy_1+n-1), \\ (1, gx_1+m, gy_1+n), (1, gx_1+m, gy_1+n+1), \\ (1, gx_1+m+1, gy_1+n-1), (1, gx_1+m+1, gy_1+n), \\ (1, gx_1+m+1, gy_1+n+1)\} \quad (28)$$

に属するセルの確率分布ベクトル $\Lambda(1, j, k, *)$ を用いて、式(2)で算出される $\gamma(v|w)$ を要素とする $W$ 次元ベクトル $\gamma = (\gamma(1|w), \gamma(2|w), \dots, \gamma(W|w))^T$ を計算する：

$$\gamma(v|w) = \prod_{(j', k') \in X(1+1, m, n) - (1, j, k)} \langle \phi(1+1, gx_1+m, gy_1+n, v|1, j', k', *) | \Lambda(1, j', k', *) \rangle \quad (29)$$

ただし、 $w$ はセル $(1, j, k)$ の状態値の一つであり、右辺の乗積は近傍系 $X(1+1, gx_1+m, gy_1+n)$ に属するセルのうち、 $(1, j, k)$ 以外のセルに渡って計算される。

【0091】ステップS1405では、セル $(1+1, gx_1+m, gy_1+n)$ と、その近傍系 $X(1+1, gx_1+m, gy_1+n)$ に属するセ

$$\sigma(1, j, k) = - \sum_w \Lambda(1, j, k, w) \sum_v \gamma(v|w) \log \{ \phi(1+1, gx_1+m, gy_1+n, v|1, j, k, w) \} \\ + \sum_w \Lambda(1, j, k, w) \sum_v \gamma(v|w) \log \{ \langle \phi(1+1, gx_1+m, gy_1+n, *) | 1, j, k, *) | \Lambda(1, j, k, *) \rangle \} \quad (30)$$

なる。

【0082】ステップS1009で、上記 $dP$ を用いてセルの確率分布ベクトルを以下のように修正する：

の階層レベル1のセル1705に対して、階層レベル2のセル1707の状態値の確率分布を表している。

【0087】更に、ステップS1402で $m$ を $-dM$ に初期化する。ここで $dM = M_0 - M_0$ である。そして、ステップS1403で $n$ を $-dN$ に初期化する。ここで $dN = N_0 - N_0$ である。

【0088】これら $m$ 及び $n$ は、図17中のセル1707を構成する階層レベル1のセルの集合として、1704、1705、及び1706のように、現在の注視点を中心にして、縦横それぞれに沿って1セル分ずつの平行移動を考慮するために設けられた操作である。

【0089】ステップS1404では、セル空間で以下の位置にあるセル、即ち図17において1704、1705、或いは1706で示されているセル、即ち以下の近傍系 $X(1+1, gx_1+m, gy_1+n)$ ：

【0090】  
【外10】

【0093】ステップS1406では、上記相互情報量  

$$B(l, j, k) = B(l, j, k) + \sigma(l, j, k)$$

【0094】ステップS1407では、(30)式で計算した相互情報量 $\sigma(l, j, k)$ 、 $(l, j, k) \in X(l+1, g_{x1}+m, g_{y1}+n)$ のうち、零でないものが存在するかどうかを检查し、存在すればステップS1408へ、そうでなければステップS1404へ戻る。

$$\beta = \beta + \gamma$$

【0096】ステップS1409で、Cの値を更新す

$$C(l, j, k) = C(l, j, k) + 1, (l, j, k) \in X(l+1, m, n)$$

ステップS1410では、 $n \leq n+1$ に更新する。ステップS1411では、 $n \leq 4M$ を超えたかどうかを检查し、超えていなければステップS1412へ進み、超えていなければステップS1404へ戻る。

【0097】ステップS1412では、 $m \leq m+1$ に更新す

$$B(l, m, n) = B(l, m, n) / C(l, m, n),$$

$$\forall m \in [-M_0, M_0] \subset Z, \forall n \in [-N_0, N_0] \subset Z$$

【0099】ステップS1415では、関数 $\lambda$ を用いて確率分布ベクトル $\beta$ を正規化する。

【0100】次に、ステップS311で実行される注視点座標の更新処理を、図18に沿って説明する。

【0101】まず、ステップS1801では、 $l$ を1に設定する。

【0102】ステップS1802では、注視点移動量を計算する。この処理を、図19のフローチャートに沿って説明する。

【0103】ステップS1901では、 $l \leq 4L$ を超えるかどうかを检查し、超えていなければ処理を終了し、そうでな

$$g_{x1} = g_{x1} + \max_x, \quad g_{y1} = g_{y1} + \max_y,$$

【0106】ステップS1904では、 $l \leq 1$ かどうかを检查し、1でなければステップS1905へ進み、1であれば処理を終了する。

【0107】ステップS1905では、 $\max_x$ 及び $\max_y$ を

$$A(l, m, n, *) = A(l, m, n, \max_x, n, \max_y, *)$$

【0108】ステップS1906では、 $(g_{x1-1}, g_{y1-1})$ を(0,0)にリセットする。

【0109】ステップS1907は、ステップS1902において $B(l, \sim, \sim)$ の中に最大値を取る零でない要素が存在しなかった場合に実行され、 $l$ を $l+1$ に更新する。

【0110】ステップS1908では、図18のステップS1802の注視点移動量の計算を実行する。このように図19で説明されている処理は再帰的である。

【0111】上記認識処理のソフトウェアによる実現例を図28から図41に示す。これらはMATLABという高級言語で書かれている。

【0112】図41は、このソフトウェアを実行するためのハードウェア構成の一例を示す図である。

【0113】同図において、4101は、入力部であり、認識対象となる画像信号を外部から受信するためのインターフェースや、ユーザが命令等を入力するためのキーボードやマウスを備えている。4102は、装置各

を用いてBを更新する：

(31)

ステップS1410へ進む。

【0095】ステップS1408では、 $\beta$ を $\gamma$ で更新する：

(32)

る：

(33)

る。ステップS1413では、 $m \leq 4M$ を超えたかどうかを检查し、超えていなければステップS1414へ進み、超えていなければステップS1403へ戻る。

【0098】ステップS1414では、多重解像度データCを用いて多重解像度データBを更新する：

(34)

ければステップS1902へ進む。

【0104】ステップS1902では、 $B(l, \sim, \sim)$ のなかに最大値を取る零でない要素が存在するかどうかを检查し、存在すればステップS1903へ、そうでなければステップS1907へ進む。

【0105】ステップS1903では、最大値に対応するインデックスをもとにして注視点位置を変更する。最大値を取る要素の、2次元空間に対応する2つのインデックスを $\max_x$ 及び $\max_y$ とすると、(17)式によって進数で記述されている注視点位置のうち、レベル $l$ に対応する要素のみが次式で変更される：

(35)

もとに、階層レベル $l-1$ に属する確率セルオートマトンのセルのセル空間における位置を平行移動させる。即ち、

(36)

部を制御し、プログラムの処理を実行するCPUである。4103は、出力部であり、認識結果を表示あるいは印刷したり、外部装置に対して、認識結果や制御信号を送信する。

【0114】4104は、ROMであり、固定データやプログラムを記憶する。4105は、RAMであり、多重解像度画像や確率オートマトンのセルの状態分布の確率分布、量子化コードブックなど、認識処理に必要なデータを記憶するためのワークエリアを有し、補助記憶装置4106からロードされるプログラムを記憶するためのプログラムエリアを有する。4106は、補助記憶装置であり、フロッピーディスクやハードディスクなどの記憶媒体を用いてプログラムやデータを記憶する。

【0115】図28から図36はsaccade6(・)という名称の主プログラムであり、与えられた2次元デジタル画像を認識する機能を持つ。確率オートマトンの構造そのものは与えられていない。セル空間の構造は図8、多

重解像度部分画像は図9に示すとおりである。

【0116】図20はコードブック及びセル間の状態値の対応表を示している。

【0117】同図において、縦横3個のマスで書かれているのが代表ベクトル、或いはセルの状態値の空間的配置、その下に書かれている番号が、それぞれ代表ベクトルに対応する符号語、或いは対応するセルの状態値である。

【0118】図13の1310で示されるコードブックが、図20の2から9までの番号を割り当てられている縦横3画素のパターンに対応する。また図17の1707で示されているセル間の写像は、図20の10から15の番号が割り当てられているパターンに対応する。

【0119】図21は、図20で番号10から13が割り当てられている状態ベクトルを2次元画像として表現したものである。また図22は同様に14と15が割り当てられている状態ベクトルを2次元画像として表現したものである。

【0120】図37から図38にはsaccade6(・)で呼び出されているサブルーチン、mutinf(・)である。mutinf(・)は相互情報量を計算するプログラムである。

【0121】図39に示されているextract2(・)と図40に示されているsubsampl(・)は、入力画像から多重解像度部分画像を抽出するサブルーチンである。

【0122】図20の状態ベクトル15を入力画像とし、画像の左上の画素の座標を(1,1)として縦の座標系を下方方向に、横の座標系を右方向に取って、注視点を(2,2)で初期化してsaccade6(・)を実行すると、図23に示すように、注視点の位置は(2,2)から(20,20)へ移動して停止する。

【0123】これら注視点に対して、状態値14及び15に対する確率Pr(14)及びPr(15)は、注視点(2,2)のときにはPr(14)=0.5、Pr(15)=0.5であり、注視点が(20,20)に移動すると、Pr(14)=0.0058、Pr(15)=0.9942となる。このように、一度の注視点移動で認識が完了したことになる。

【0124】本実施形態による注視点移動方式のかわりに画像の特徴量のみに基づく方式を用いたとすれば、少なくとも10回の注視点移動が必要となる。

【0125】図2のステップS205で実行される学習処理について、図24、図25及び図26に沿って説明する。図24に示すように、学習処理はステップS2401の符号化処理とステップS2402の対応付け処理の2段階より構成される。

【0126】図25は符号化処理の流れ図である。

【0127】まず、ステップS2501では入力画像Iを多重解像度画像に変換する。ただし入力画像Iの縦横の画素数は、3の(l+1)乗であるとする。

【0128】ステップS2502では、lを0に設定する。

【0129】ステップS2503では、多重解像度画像のうちの解像度レベルlの画像を部分画像に分割する。本実施例では、縦横3画素で互いに重なり合わない部分画像に分割する。

【0130】以下、部分画像を、行列の表記に従い、2次元のインデックス(m,n)で表現する。

【0131】ステップS2504では、nを1に設定する。ステップS2505では、mを1に設定する。

【0132】ステップS2506では、部分画像(m,n)がベクトル量子化のコードブックに代表ベクトルとして登録されているかどうかを検査し、登録されていないステップS2508へ、未登録ならばステップS2507へ進む。

【0133】ステップS2507では、上記部分画像に適当な符号語を割り当ててコードブックに登録する。

【0134】ステップS2508では、それぞれの部分画像を、対応する符号語で置き換える。この置き換え操作によって部分画像の画素数は、縦横それぞれ1/3に縮小する。

【0135】ステップS2509では、mをm+1に更新する。ステップS2510では、mが3を超えているかどうかを検査し、超えていなければステップS2711へ、そうでなければステップS2506へ進む。

【0136】ステップS2511では、nをn+1に更新する。ステップS2512では、nが3を超えているかどうかを検査し、超えていなければステップS2513へ、そうでなければステップS2508へ進む。

【0137】ステップS2513では、lをl+1に更新する。ステップS2514では、lが4を超えていないかを検査し、超えていなければ終了し、そうでなければステップS2503へ戻る。

【0138】以上の処理によって、多重解像度画像は、各解像度レベルでの画素数が縦横それぞれ1/3に縮小された多重解像度符号語データに変換される。

【0139】また、コードブックに登録されていない代表ベクトルを登録することによって、新しいデータに対応することができ。

【0140】上記処理で得られるコードブックは、前述した本実施形態の認識処理において利用される。

【0141】図26は、上記多重解像度符号語データを用いて、レベル間の対応関係の抽出と登録を行う処理のフローチャートである。

【0142】以下、多重解像度符号語データを解像度レベルl、2次元離散座標(m,n)の3つのインデックスを用いて、Y(l,m,n)で表現する。

【0143】ステップS2601では、lを1に設定する。ステップS2602では、nを1に設定する。ステップS2603では、mを1に設定する。

【0144】ステップS2604では、符号語Y(l,m,n)と、以下の符号語の組：

$\{Y(-1, 3m-1, 3n-1), Y(-1, 3m-1, 3n), Y(-1, 3m-1, 3n+1)\}$

$\{Y(-1, 3m, 3n-1), Y(-1, 3m, 3n), Y(-1, 3m, 3n+1)\}$

$\{Y(-1, 3m+1, 3n-1), Y(-1, 3m+1, 3n), Y(-1, 3m+1, 3n+1)\}$

との対応関係を抽出する。

【0145】ステップS2605では、上記対応関係が、コードブック或いは対応表に登録されているかどうかを検査し、登録されていればステップS2607へ、そうでなければステップS2606へ進む。

【0146】ステップS2606では、対応表に、上記対応関係を登録する。

【0147】ステップS2607では、 $m$ を $m+1$ に更新する。ステップS2608では、 $m$ が3を超えているかどうかを検査し、超えていなければステップS2609へ、そうでなければステップS2604へ進む。

【0148】ステップS2609では、 $n$ を $n+1$ に更新する。ステップS2610では、 $m$ が3を超えているかどうかを検査し、超えていなければステップS2611へ、そうでなければステップS2603へ進む。

【0149】ステップS2611では、 $l$ を $l+1$ に更新する。ステップS2612では、 $l$ が $l$ を超えていないかを検査し、超えていければ終了し、そうでなければステップS2602へ戻る。

【0150】上記処理によって得られる対応表は、前述した本実施形態の認識処理において利用される。

【0151】(実施形態2)図27に実施形態2の構成図を示す。

【0152】実施形態2は、監視システム、ロボットの画像入力システム等への本発明の応用例であり、図中、2701は信号処理装置、2702は記憶装置、2703は出力装置、2704は光学系、2705は2次元アレイセンサ、2706は第1の筐体、2707は第2の筐体、2708は移動装置である。

【0153】信号処理装置2701は、光学系2704を経て、2次元アレイセンサ2705でサンプリングされた入力信号を受信し、記憶装置2702に格納されたデータを用いて入力信号を処理し、所望の出力を出力装置2703へ送信し、光学系2704の入力パラメータ(光軸の方位や位置など)を制御するための信号を移動装置2708に送信する。

【0154】実施形態2で想定している応用例では、入力すべき画像が入力システムのどの方向に存在するかは、予め与えられていない。従って、システムが実行すべきタスクに必要な情報が存在する方位を推定し、その方向に入力装置の光軸を一致させ、その情報を取得するために必要な解像度で撮像するように、入力パラメータを制御することが要求される。

【0155】また、撮像系に対しては、高解像度で信号を取得し、更に広範囲な領域をも見渡せることが要求さ

れる。限られたセンサ数で、この相反する仕様を満足するために、撮像系の空間解像度は非一様とする。実施形態1で説明した多重解像度部分画像4は、このような非一様サンプリングによって得られる入力画像の1例である。

【0156】実施形態2は、このような入力装置を想定するものである。

【0157】<光学系2704>光学系2704は、外部から入射してくる光線を屈折させて、後述の2次元アレイセンサ2705上に結像させるためのものである。高解像度と広範囲を両立させるレンズとして、例えば魚眼レンズがある。魚眼レンズは、その写像関数によって4種類に分類される。

【0158】<2次元アレイセンサ2705>2次元アレイセンサ2705は、前記光学系2704によって結像した画像を2次元デジタル画像としてサンプリングする。ここで、2次元デジタル画像とは、2次元空間座標を離散化し、更にそれぞれのサンプリング点での画素値を量子化したものをいう。

【0159】光学系2704として魚眼レンズを採用し、その像を2次元アレイセンサでサンプリングして得られたデジタル画像は、擬似的に多重解像度部分画像になる。そこで、以下の説明では、このデジタル画像を実施形態1での記述と同様にDと書く。Dは解像度レベル1、2次元離散空間座標 $(m, n)$ の3つのインデックスで画素位置を表す。例えば解像度レベル1、2次元離散空間座標 $(m, n)$ に位置する画素値は $D(1, m, n)$ と書く。

【0160】<第1の筐体2706>第1の筐体2706は前記光学系2704と前記2次元アレイセンサ2705とを固定する。

【0161】<第2の筐体2707>第2の筐体2707は、後述の移動装置2708を介して、第1の筐体2706を支持するものである。

【0162】<移動装置2708>移動装置2708は、第1の筐体2706を、第2の筐体2707に対して2次元的に相対移動させるものである。2次元移動量は、後述の信号処理装置2701からの制御信号で制御される。

【0163】<記憶装置2702>記憶装置2702は、後述する信号処理装置2701の処理に利用される信号を記憶しておく装置であり、信号処理装置2701によって信号が書き込まれ、或いは読み出される。

【0164】<出力装置2703>出力装置2703は、後述する信号処理装置2701からの出力を表示或いは印刷するものである。

【0165】<原稿置き台2709>原稿置き台2709は、後述する信号処理装置2701で認識する画像或いはパターンが描かれている原稿2710を支持するための台である。原稿に描かれている画像或いはパターンが、原稿置き台2709を通して光学系に入力される配

置になっているときには、原稿置き台2709は透明でなければならない。

【0166】<原稿2710>後述する信号処理装置2701で認識される対象となる画像或いはパターンが描かれている原稿である。

【0167】<信号処理装置2701>信号処理装置2701は、2次元アレイセンサ2705から送信されてきたデジタル画像を認識する装置であり、認識処理を実行するために記憶装置2702に格納された信号を読み出し、或いは信号を書き込み、また移動装置2708に2次元移動量を制御するための制御信号を送信する。また認識結果や、何らかのエラーが発生した場合のメッセージを出力装置2703に出力する。

【0168】信号処理装置2701の処理は、実施形態1で説明した全ての処理を含み、実施形態1の説明文中の注視点位置が、実施形態2における光学系の光軸に相当する。また、実施形態1の説明文中の入力画像Iは、実施形態2における原稿2710に相当する。

【0169】

【発明の効果】以上説明したように、本発明によれば、非一様サンプリングによりサンプリングした信号を用いてパターン認識を行なう際に、パターンをより正確に認識するために、サンプリング位置を高速に修正することができる。

【0170】また、本発明によれば、認識対象パターンの存在する方位が不明な場合でも、パターンを高速に捕捉することができるので、移動物体に設置するために好適なパターン認識装置を容易に構成することができる。

【0171】また、本発明によれば、従来方式に比べて、比較にならないほど少ないセンサ数で構成されたアレイセンサで、パターン認識処理が構成できる。例えば、実施形態1で説明した例では、従来方式で3<sup>11</sup>個のセンサが必要なのに対して、実施形態1では、(31)<sup>2</sup>個のセンサで十分である。

【図面の簡単な説明】

【図1】実施形態1のパターン認識装置のブロック構成図である。

【図2】信号処理装置の処理のフローチャートである。

【図3】認識処理のフローチャートである。

【図4】認識処理のフローチャートである。

【図5】多重解像度部分画像の抽出処理のフローチャートである。

【図6】解像度レベル1の画像の抽出処理のフローチャートである。

【図7】解像度レベル1の画像の抽出処理のフローチャートである。

【図8】確率セルオートマトンのセル空間の説明図である。

【図9】多重解像度部分画像の説明図である。

【図10】確率分布の更新処理のフローチャートであ

る。

【図11】確率分布の更新処理のフローチャートである。

【図12】確率分布の更新処理のフローチャートである。

【図13】確率分布の更新処理の説明図である。

【図14】相互情報量の算出処理のフローチャートである。

【図15】相互情報量の算出処理のフローチャートである。

【図16】相互情報量の算出処理のフローチャートである。

【図17】相互情報量の算出処理におけるデータ間の関係を示す図である。

【図18】注視点座標の更新処理のフローチャートである。

【図19】注視点移動の計算処理のフローチャートである。

【図20】符号化処理のコードブック及びセルの状態間の写像関係の例を示す図である。

【図21】代表ベクトルに対する2次元画像である。

【図22】代表ベクトルに対する2次元画像である。

【図23】注視点移動を示した図である。

【図24】コードブックやセル間写像の学習のフローチャートである。

【図25】符号化処理のフローチャートである。

【図26】対応づけ処理のフローチャートである。

【図27】実施形態2のブロック構成図である。

【図28】2次元デジタル画像認識処理ソフトウェアの主プログラムを表わす図である。

【図29】2次元デジタル画像認識処理ソフトウェアの主プログラムを表わす図である。

【図30】2次元デジタル画像認識処理ソフトウェアの主プログラムを表わす図である。

【図31】2次元デジタル画像認識処理ソフトウェアの主プログラムを表わす図である。

【図32】2次元デジタル画像認識処理ソフトウェアの主プログラムを表わす図である。

【図33】2次元デジタル画像認識処理ソフトウェアの主プログラムを表わす図である。

【図34】2次元デジタル画像認識処理ソフトウェアの主プログラムを表わす図である。

【図35】2次元デジタル画像認識処理ソフトウェアの主プログラムを表わす図である。

【図36】2次元デジタル画像認識処理ソフトウェアの主プログラムを表わす図である。

【図37】相互情報量を計算するプログラムを表わす図である。

【図38】相互情報量を計算するプログラムを表わす図である。

【図39】多重解像度部分画像を抽出するプログラムを表わす図である。

【図40】多重解像度部分画像を抽出するプログラムを表わす図である。

【図41】ハードウェア構成の例を示す図である。

【符号の説明】

101 入出力装置

102、2701 信号処理装置

103、2702 記憶装置

2703 出力装置

2704 光学系

2705 2次元アレイセンサ

2706 第1の筐体

2707 第2の筐体

2708 移動装置

2709 原稿置き台

2710 原稿

4101 入力部

4102 CPU

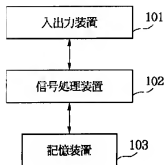
4103 出力部

4104 ROM

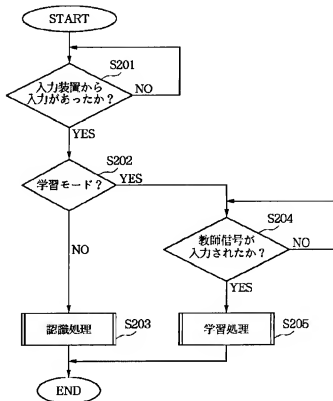
4105 RAM

4106 補助記憶装置

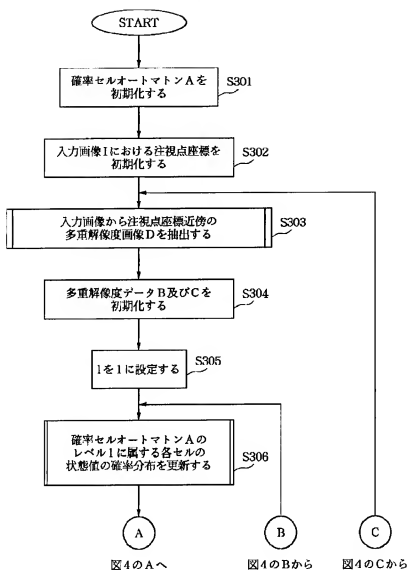
【図1】



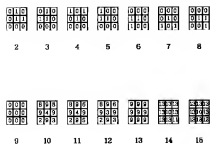
【図2】



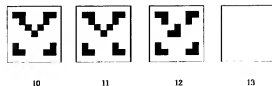
【図3】



【図20】

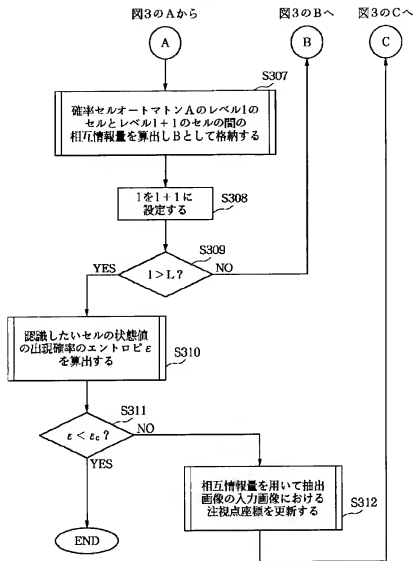


【図21】





【図4】



【図40】

Subsaml.m

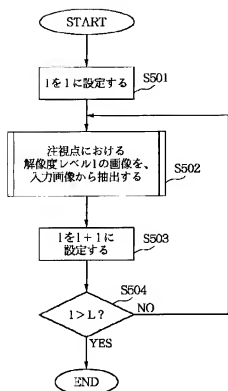
```

function Y=subsaml(X,s)

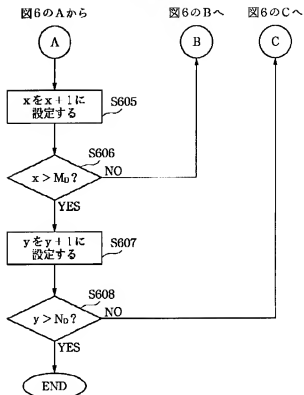
[M,N]=size(X);
for m=1:fix(M/s)
    for n=1:fix(N/s)
        Y(m,n)=X((m-1)*s+round(s/2):(m-1)*s+round(s/2)):
    end
end
end

```

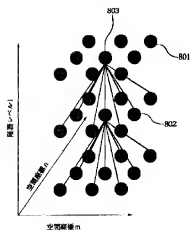
【図5】



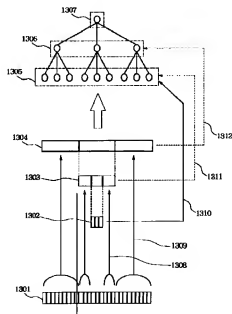
【図7】



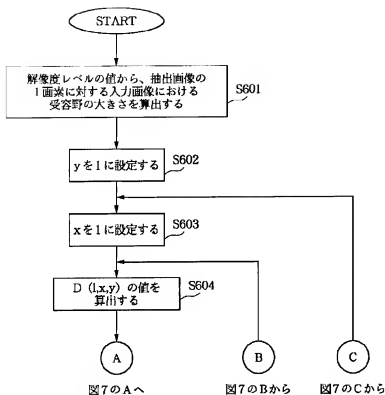
【図8】



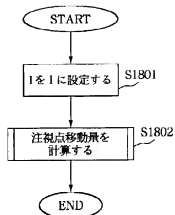
【図13】



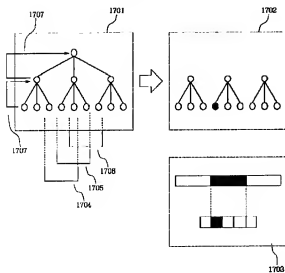
【図6】



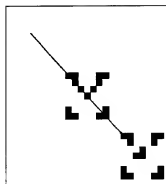
【図18】



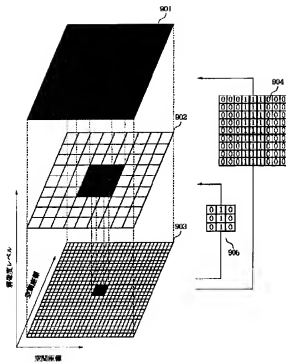
【図17】



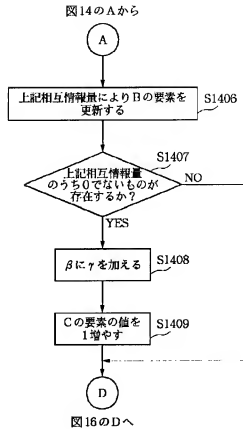
【図23】



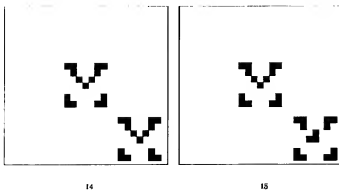
【図9】



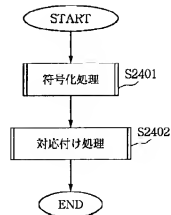
【図15】



【図22】



【図24】



【図10】

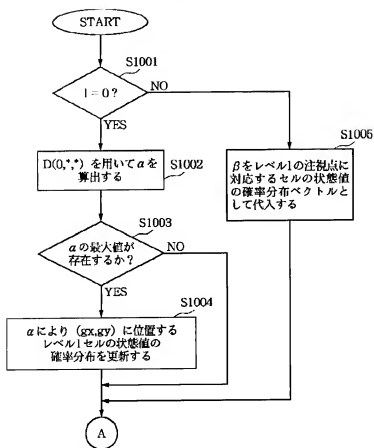
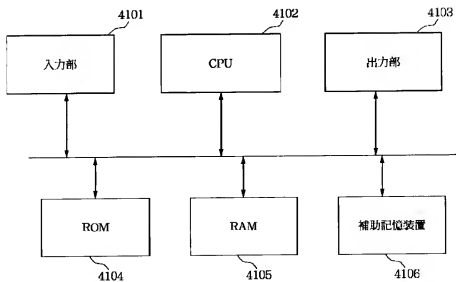
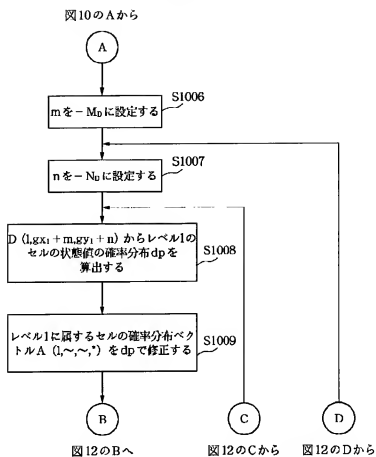


図 11 の A へ

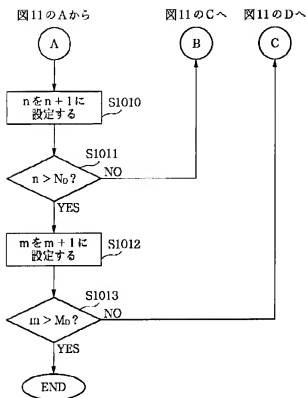
【図11】



【図11】



【図12】



【図36】

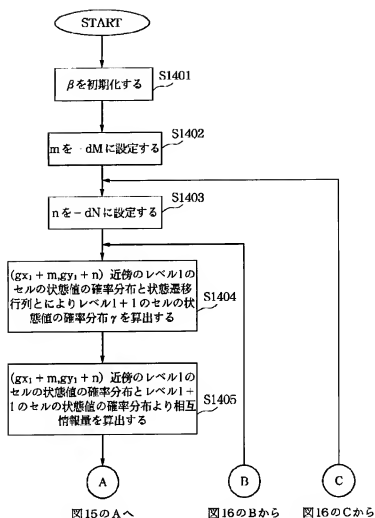
Saccade6.m

```

for mm=1:sx1
  for nn=1:syl
    if real((mm+i*nn)-dgaze/3)>=1 & real((mm+i*nn)-dgaze/3)<=sx1
      if imag((mm+i*nn)-dgaze/3)>=1 & imag((mm+i*nn)-dgaze/3)<=syl
        nSTM1(mm,nn)-STM1(real((mm+i*nn)-dgaze/3),imag((mm+i*nn)-dgaze/3));
      end
    end
  end
end
STM1=nSTM1;
STAGE1=0;
else
  dgaze=(dx1-(3+i*3))*3^1;
  PNT1-CTR1+dx1-(3+i*3);
end
gaze=init_gaze+(PNT2-CTR2)*3^2+(PNT1-CTR1)*3^1;
seq_gaze=[seq_gaze;gaze];
gaze
end
end
% end of list

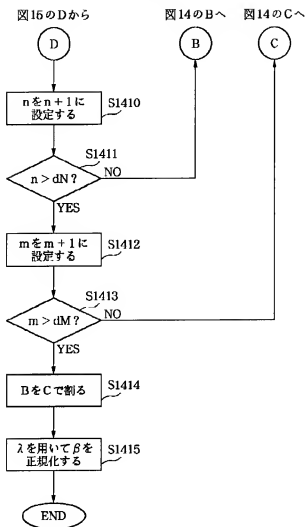
```

【図14】

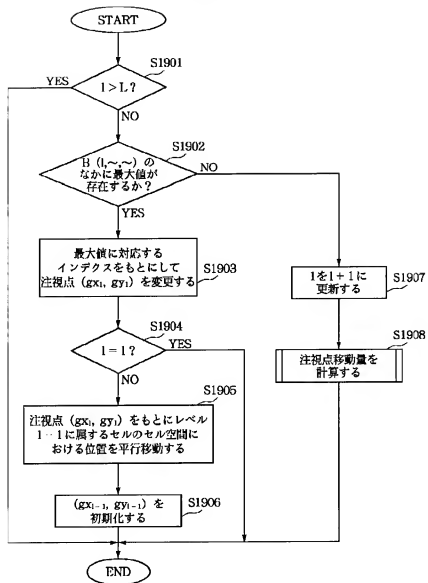




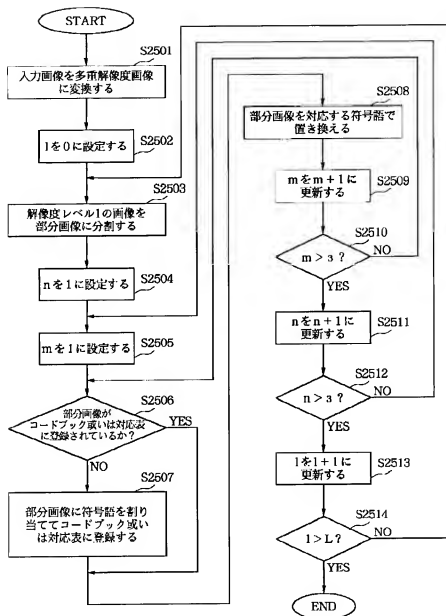
【図16】



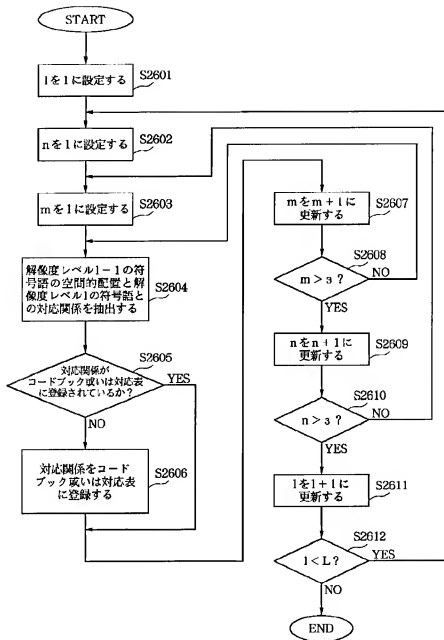
【図19】



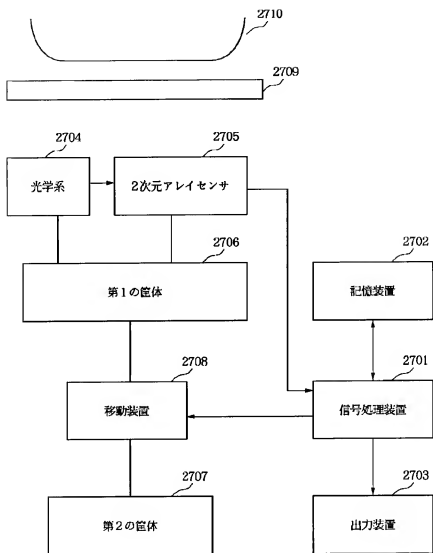
【図25】



【図26】



【図27】



【図28】

## Saccade6.m

```

% includes spatial hierarchical structure.

function[seq_gaze, seq_ycl, seq_yc2]=saccade6(itr, init_gaze, crt)

VQMATO=[-1 -1 -1 +1 +1 -1 -1 +1 -1;      %9
         -1 +1 -1 +1 +1 -1 -1 -1 -1;      %10
         +1 -1 -1 -1 +1 -1 +1 -1 -1;      %14
         +1 +1 -1 -1 +1 -1 -1 -1 -1;      %15
         -1 +1 +1 -1 +1 -1 -1 -1 -1;      %16
         -1 -1 +1 -1 +1 -1 -1 -1 +1;      %17
         -1 -1 -1 -1 +1 -1 -1 +1 +1;      %18
         -1 -1 -1 -1 -1 -1 -1 -1 -1];      %19

NUNOO=8;
[M, N]=size(VQMATO);
VQMATO=(VQMATO./(ones(M, 1)*sqrt(sum(VQMATO.^2))));

VQMATC1=[0 0 0 0 0 1 0;      %1
          0 0 0 0 0 0 1;
          1 0 0 0 0 0 0;
          0 0 0 0 0 0 1;
          0 0 1 0 0 0 0;
          0 0 0 0 0 0 1;
          0 0 0 1 0 0 0;
          0 0 0 0 0 0 1;
          0 1 0 0 0 0 0;
          0 0 0 0 0 1 0;      %2
          0 0 0 0 0 0 1;
          1 0 0 0 0 0 0;
          0 0 0 0 0 0 1;
          0 0 1 0 0 0 0;
          0 0 0 0 0 0 1;
          0 0 0 1 0 0 0;
          0 0 0 0 0 0 1;
          0 0 1 0 0 0 0;
          0 0 0 0 0 1 0;      %3
          0 0 0 0 0 0 1;
          1 0 0 0 0 0 0;
          0 0 0 0 0 0 1;

```







## 【図31】

## Saccade6.m

```

oy=(sy-3)/2;
FLG3=0;
gaze=init_gaze;
Level=3;

INPUT= 1/3*ones(27);
INPUT=impatch2(hs21, INPUT, 14+i*14, 1, 0);
INPUT=impatch2(hs23, INPUT, 23+i*23, 1, 0);

STM1=1/NUM00*ones(NUM00, 1)*ones(1, sx1*sy1);
STM2=1/NUM01*ones(NUM01, 1)*ones(1, sx1*sy1);
MIF1=zeros(sx, sy);
MIF2=zeros(sx, sy);
CTR1=round(sx1/2)+i*round(sy1/2);
CTR2=round(sx1/2)+i*round(sy1/2);
PNT1=CTR1;
PNT2=CTR2;
seq_gaze=init_gaze;
seq_ycl=[];
seq_yc2=[];
STAGE1=0;
STAGE2=0;

for t=1:itr
    x0=extract2(INPUT, 0, gaze, 3, 3);
    x1=extract2(INPUT, 0, gaze, 9, 9);
    x1=subsaml(filter2(fh1, x1, 'same'), 3);
    x2=extract2(INPUT, 0, gaze, 27, 27);
    x2=subsaml(filter2(fh2, x2, 'same'), 9);
    x2=mat2vec2(x2, 3, 3, 0);
    x1=mat2vec2(x1, 3, 3, 0);
    x0=mat2vec2(x0, 3, 3, 0);
    if sum(x0)~=0
        x0=x0/sqrt(sum(x0.^2));
    end

    ys0=VQMAT0'*x0;
    [mmax, mmy] = detmax(ys0);

```

【図32】

```

Saccade6, m
if max~-1:~(r1) % this assumes any level-0 pattern can be classified by one level-0 observation.
    STM(:, (imag(PNTI)-1)*sxl+real(PNTI))=zeros(NUM0, 1);
    STM(real(max), (imag(PNTI)-1)*sxl+real(PNTI))=1;
else
    end
for m=1:1
    for n=1:1
        dys=zeros(NUM0, 1);
        for ind=1:NUM0
            if xl*(m-2)-(n-1)*3 >= VQWATS(indx, 1)+0, 99 & xl*(m-2)-(n-1)*3 <= VQWATS(indx, 1)+1, 01
                dys(indx, 1)=1;
            end
        end
        if sum(dys)~=0
            dys=dys/sum(dys);
        else
            dys=ones(NUM0, 1);
        end
        STM(:, (imag(PNTI)-n-1)*sxl+real(PNTI)-m)-STM(:, (imag(PNTI)+n-1)*sxl+real(PNTI)-m)+dys;
        sSTM=sum(STM(:, (imag(PNTI)-n-1)*sxl+real(PNTI)-m));
        if sSTM~=0
            STM(:, (imag(PNTI)+n-1)*sxl+real(PNTI)+m)-STM(:, (imag(PNTI)+n-1)*sxl+real(PNTI)-m)/sSTM;
        end
    end
end

MFI=zeros(ox, oy);
CQMAT=zeros(ox, oy);
yyel=zeros(NUM0, 1);
for m=1:3
    for nn=1:3
        sml=[];
        for mm=0:2
            for nnn=0:2
                STM(:, (nn-1+mm-1)*sxl+nn-1+mm)];
            end
        end
        yel=prod(sum(mat2vec2(VQWATCI(:, 1), NUM0, 3+3, 1), *sml));
    end
end

```

## 【図33】

```

                                Saccade6.m
ycl=[ycl:prod(sum(mat2vec2(VQMATC1(:,2),NUM00,3*3,1).*stml'))];
ycl=ycl:prod(sum(mat2vec2(VQMATC1(:,3),NUM00,3*3,1).*stml'))];
ycl=ycl:prod(sum(mat2vec2(VQMATC1(:,4),NUM00,3*3,1).*stml'))];
stml=mat2vec2(stml,NUM00,3*3,0);
MIF1(mm:mm+mmm,nn::nn+nnn)=MIF1(mm:mm+mmm,nn:nn+nnn)+mutinfl(VQMATC1,ycl,stml,3,3);
if sum(sum(abs(MIF1)))>eps
    yycl=yycl+ycl;
    CMBMAT(mm:mm+mmm,nn::nn+nnn)=CMBMAT(mm:mm+mmm,nn:nn+nnn)+ones(3);
end
end
end
for m=1:sx
    for n=1:sy
        if CMBMAT(m,n)==0
            if CMBMAT(m,n)=1;
            end
        end
    end
end
MIF1=MIF1./CMBMAT;

if sum(yycl)>eps
    yycl=yycl/sum(yycl);
else
    yycl=ones(NUM01,1)/NUM01;
end
[mmx,mmv]=detmax(yycl);
% if mmv>=1-eps & mmx=-1+i*(-1)
if mmv>=crt*1 & mmx=-1+i*(-1)
    STAGE1=1;
else
    end
STM2(:,(imag(PNT2)-1)*sx+real(PNT2))=yycl;
for m=1:l
    for n=1:l
        dys=zeros(NUM01,1);
        for indx=1:NUM01
            if x2((m/2)+(n/1)*3)>=VQMAT2S2(indx,1)*0.99 & x2((m/2)+(n/1)*3)<=VQMAT2S2(indx,1)*1.01

```

【図 3 4】

Saccade6. m

```

        dys(indx,1)-1;
    end
end
if sum(dys)~=0
    dys=dys/sum(dys);
else
    dys=ones(NUM01,1);
end
STM2(:,(imag(PNT2)+n-1)*sx1+real(PNT2)+m)=STM2(:,(imag(PNT2)+n-1)*sx1+real(PNT2)+m).*dys;
sSTM2=sum(STM2(:,(imag(PNT2)+n-1)*sx1+real(PNT2)+m));
if sSTM2~=0
    STM2(:,(imag(PNT2)+n-1)*sx1+real(PNT2)+m)=STM2(:,(imag(PNT2)+n-1)*sx1+real(PNT2)+m)/sSTM2;
end
end
end

MIF2=zeros(sx,sy);
CMRMAT=zeros(sx,sy);
yyc2=0;
for mm=1:3
    for nn=1:3
        stm2=[];
        for mmm=0:2
            for nnn=0:2
                stm2=[stm2 STM2(:,(mm+1+mmm-1)*sx1+nn+1+nnn)];
            end
        end
        yyc2=prod(sum(mat2vec2(VQWATC2(:,1),NUM01,3*3,1).*stm2'));
        yyc2=lyc2*prod(sum(mat2vec2(VQWATC2(:,2),NUM01,3*3,1).*stm2'));
        stm2=mat2vec2(stm2,NUM01,3*3,0);
        MIF2(mn:mm+mmm,nn:nn+nnn)=MIF2(mn:mm+mmm,nn:nn+nnn)+mutinf1(VQWATC2,yc2,stm2,3,3);
        if sum(sum(abs(MIF1)))>eps
            yyc2=yyc2+yc2;
            CMRMAT(mn:mm+mmm,nn:nn+nnn)=CMRMAT(mn:mm+mmm,nn:nn+nnn)+ones(3);
        end
    end
end
end
for m=1:sx

```

【図35】

Saccade6.m

```

for n=1:sy
    if CMBMAT(m,n)==0
        CMBMAT(m,n)=1;
    end
end
end
MIF2=MIF2./CMBMAT;

if sum(yyc2)>eps
    yyc2=yyc2/sum(yyc2);
else
    yyc2=ones(NUM02,1)/NUM02;
end
[mmx,mmy]=detmax(yyc2);
% if mmy>-1-eps & mmx==-1+i*(-1)
if mmy>-crt*1 & mmx==-1+i*(-1)
    STAGE2=1;
else
    end

seq_yc1=[seq_yc1 yyc1];
seq_yc2=[seq_yc2 yyc2];
if STAGE2==1
    break;
else
    [dx2,dy2]=detmax(MIF2);
    [dx1,dy1]=detmax(MIF1);
    if dx1==-1+i*(-1) | STAGE1==1
        PNT1=CTR1;
        if dx2==-1+i*(-1)
            mat2vec2(x2,3,3,1);
            [dgazo,dy1]=dotmax(abs(x2));
            PNT2=CTR2;
        else
            dgaze=(dx2-(3+i*3))*3^2;
            PNT2=CTR2+dx2-(3+i*3);
        end
        nSTM1=1/NUM00*ones(NUM00,1)*ones(1,sx1*sy1);
    end
end

```

【図37】

```

Matinf.m

function mif=matinf(MAT,Y,X,ax,ay)

pY=Y;

nif=zeros(ax,ay);
SMAT=sum(MAT); % summing every row vector
[N,N]=size(SMAT);
for n=1:N
    if SMAT(1,n)~=0
        SMAT(1,n)=1;
    else
        end
end
end
[N,N]=size(MAT);
PMAT=MAT./SMAT*ones(1,N); % conditional probability of X for Y

[XM,yM]=size(X);
[mX,yM]=size(Y);
[XM,yM]=size(Y);
for m=1:ax
    for n=1:ay
        for indX=1:num_X
            px=X(((n-1)*ay+m-1)*num_X+indX,1);
            entry=0;
            for indY=1:yM
                py=PMAT(((n-1)*ay+m-1)*num_X+indX,indY);
                if py==0
                    iy=0;
                else
                    iy=py*log(py);
                end
                for mm=1:ax
                    for mn=1:ay
                        if m==mm & n==mn
                            else
                                iy=iy+MAT(((n-1)*ay+mm-1)*num_X+1;((mn-1)*ay+mn)*num_X,indY)*X(((n-1)*ay+mm-1)*num_X+1;((mn-1)*ay+mm)*num_X,1);
                            end
                        end
                    end
                end
            end
        end
    end
end

```

【図38】

```

Mu:inf,m
end
end
enty=enty+iy;
end
mif(m,n)=mif(m,n)-px*enty;
end
for indY=1:YM
    py=0;
    for indX=1:num_X
        end
        py=py+MAT(((n-1)*ay+n-1)+num_X-indX,indY)*X(((n-1)*ay+n-1)+num_X+incX,1);
    end
    if py==0
        iy=0;
    else
        iy=-py*log(py);
    end
    for mm=1:ax
        for nr=1:ay
            if m==mm & n==nn
                else
                    iy=iy+MAT(((nn-1)+ay+mm-1)+num_X+1:((nn-1)+ay+nn)+num_X,indY)*X(((nn-1)+ay-
                    num_X,1);
                end
            end
        end
    end
    mif(m,n)=mif(m,n)+iy;
end
end
end
% end of list

```

## 【図39】

## Extract2.m

```

% extract2 maps a given image into subspace of the multi-resolution representation.
% Y - extract2(X,L,a,sx,sy)
% Y: resulting image.
% X: an image.
% L: a MRR level of interest.
% a: gaze position in the complex plane.
% sx: x-size of Y.
% sy: y-size of Y.
function Y = extract2(X,L,a,sx,sy)
[M,N] = size(X);
XOrg=round(M/2)+i*round(N/2);
sOrg=round(sx/2)+i*round(sy/2);
LOrg=round(3*L/2)+i*round(3*L/2);
W=zeros(sx+2,sy+2);
count=0;
for y=1:sx+2
    for x=1:sx+2
        cc=a*((x+iy)-sOrg-(1+i))*3^L;
        for dy=1:3^L
            for dx=1:3^L
                ccc=cc+((dx+idy)-LOrg);
                if (real(ccc)>=1) & (real(ccc)<=M) & (imag(ccc)>=1) & (imag(ccc)<=N)
                    W(x,y) = W(x,y) + X(real(ccc),imag(ccc))*3^(-2*L);
                else
                    end
            end
        end
    end
end
Y=W(2:sx+1,2:sy+1);
%Y=filter2(fspecial('laplacian',0.5),W,'valid');
% kernel=[0 0 0;0 1 0;0 0 0];
% Y=filter2(kernel,W,'valid');
% Y=Y.*3^(-2*L);

```